INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Department of Chemical Engineering

CH5350 : Applied Time Series Analysis
Solutions to Assignment #4

Marks Distribution

	1	2	3	4	5	6
(a)/(i)	10	20	10	8	5	20
(b)/(ii)	-	-	10	8	5	-
	-	-	-	4 (Concluding remarks)	-	-

1.

The p.s.d of a stationary process is given by

$$\gamma_{xx}(f) = \frac{1.44}{1.16 - 0.8\cos(2\pi f)}$$
$$\gamma_{xx}(\omega) = \frac{1.44}{2\pi}\frac{1}{1.16 - 0.8\cos\omega}$$
$$|H(e^{-j\omega})|^2 = \frac{1}{1.16 - 0.8\cos\omega} \Rightarrow H(q^{-1}) = \frac{1}{1 - 0.4q^{-1}}$$

It is an AR(1) process $d_1 = 0.4$ and $\sigma_e^2 = 1.44$
$\sigma[0] = \frac{\sigma_e^2}{1-d_1^2} \quad \Rightarrow \quad \sigma[0] = 1.7143$
$\sigma[1] = (-d_1)\sigma[0] \quad \Rightarrow \quad \sigma[1] = 0.6857$
Similarly $\sigma[2] = 0.2743 \;\; \sigma[3] = 0.1097 \;\; \sigma[4] = 0.0438 \;\; \sigma[5] = 0.0175$

Verifying the result with the inverse Fourier transform in R

```
> f <- seq(from = -0.5, to = 0.5, by = 0.01)
> psd <- 1.44/(1.16-0.8*cos(2*pi*f))
> invft <- abs(ifft(psd))
> invft[1:5]
[1] 1.70458679 0.68359278 0.27712131 0.11133245 0.04523069
```

2.

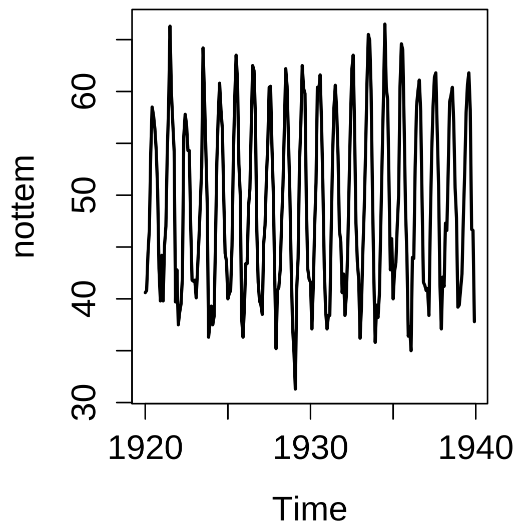
The R-code for modelling nottem data is shown below

```
# Loading the data
data('nottem')
xk=ts(nottem)
plot(xk,type='l')
xk=xk-mean(xk)
k=seq(0,239,1)
spectrum(xk)
# Period is 12 samples
mod_per <- lm (xk ~ cos(2*pi*k/12) + sin(2*pi*k/12) - 1)
# Extracting residuals
resk=mod_per$residuals
plot(resk,type='l')
acf(resk)
# Acf in significant at only one lag. Hence AR(1) model is fit for the data
mod_ar=arima(resk,c(1,0,0))
acf(mod_ar$residuals)
pacf(mod_ar$residuals)
```

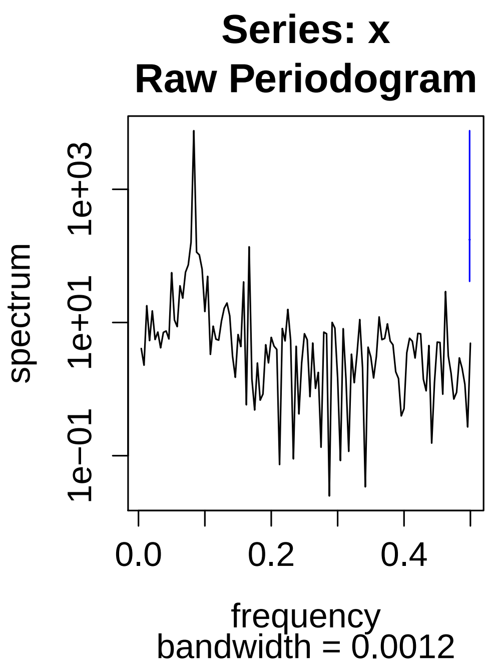
The data is shown in Figure 1a. The periodogram of the data is shown in Figure 1b. The data is periodic with a period of 12 samples. To remove periodicities the data is fitted with

$$x[k] = A\cos(2*pi*k/12) + B\sin(2*pi*k/12)$$

The periodic signal is obtained as



(a) nottem Data



(b) Spectrum of data

```
Call:
lm(formula = xk ~ cos(2 * pi * K/12) + sin(2 * pi * k/12) - 1)

Coefficients:
cos(2 * pi * K/12)   sin(2 * pi * k/12)
-11.473                -1.391
```

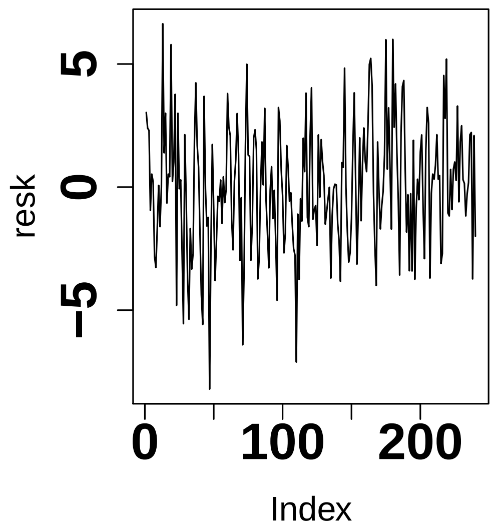
The residuals are shown in Figure 1a. The acf plot of the residuals is shown in the Figure 1b. From acf plot of residuals AR(1) model is chosen to fit for the process. The model is given by Call: arima(x = resk, order = c(1, 0, 0))

Coefficients: ar1 0.2655 (±0.064) s.e. 0.0623

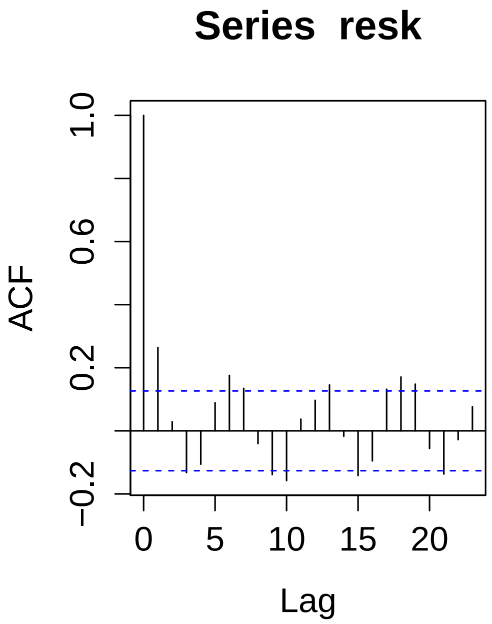
The acf and pacf plot of residuals is shown in Figures The final model is given by

$$x[k] = -11.473cos(2 * pi * K/12) - 1.391sin(2 * pi * k/12) + v[k]$$

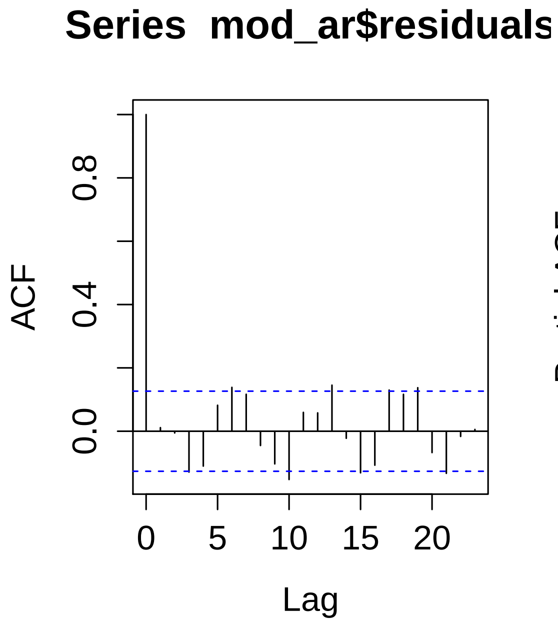
$$\text{where} \quad v[k] = 0.2655v[k-1] + e[k]$$



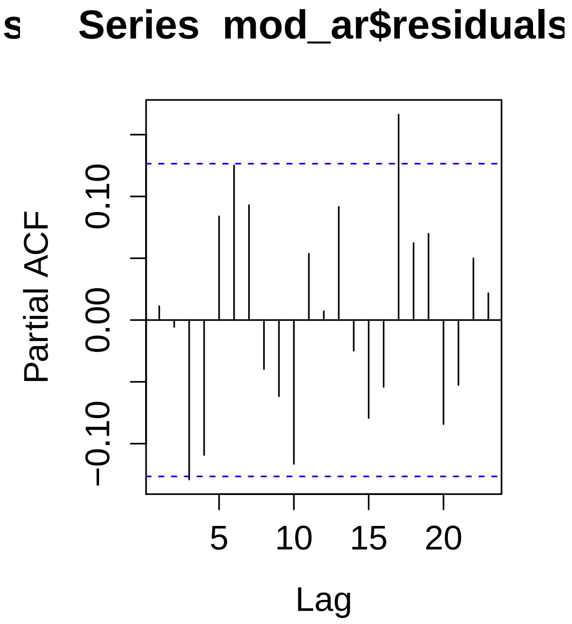
(a) Residual plot



(b) ACF plot of residuals



(a) ACF plot of residuals



(b) PACF plot of residuals

3.

(a).

The data is shown in Figure 1

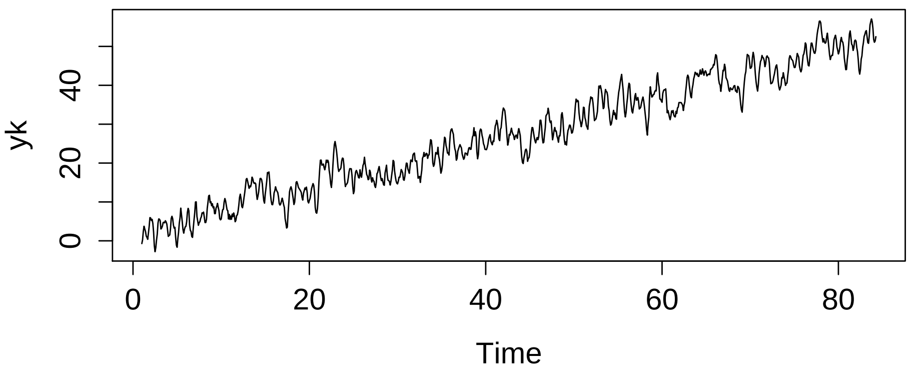


Figure 1: Cardiovascular data

Clearly, the data has some seasonal component. The data is decomposed using stl command and is shown in Figure 3

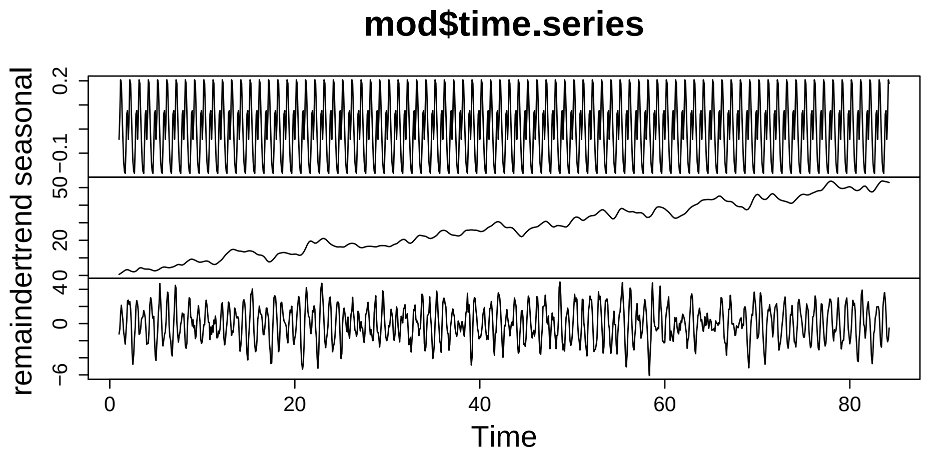


Figure 2: SARIMA data

The acf of remainder series is shown in Figure 4

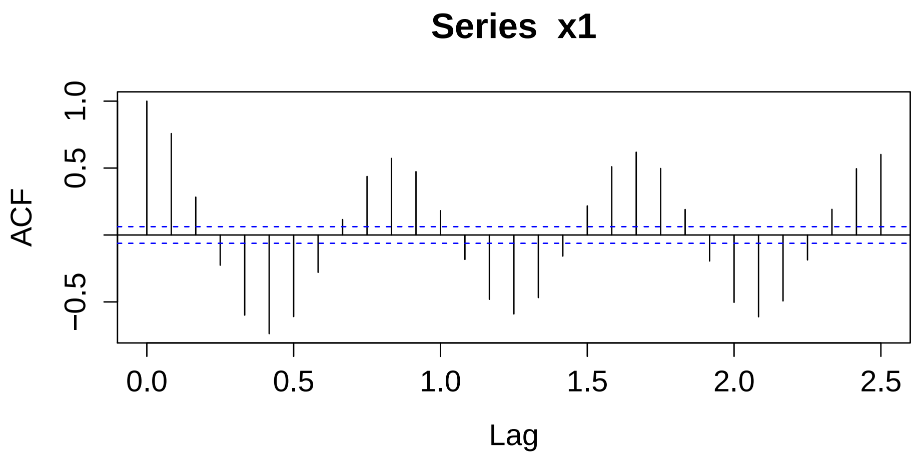


Figure 3: decomposed data

Clearly, the signal is periodic with a period of 10 samples. Hence a sum of cosine and sine signal is fitted using lm command. The acf of residuals is shown in Figure 5

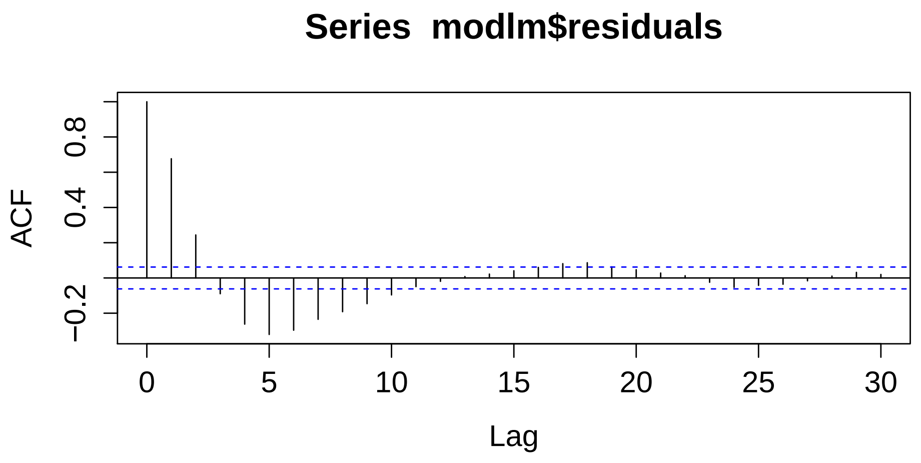


Figure 4: ACF of residuals

The pacf of residuals is shown in Figure 6

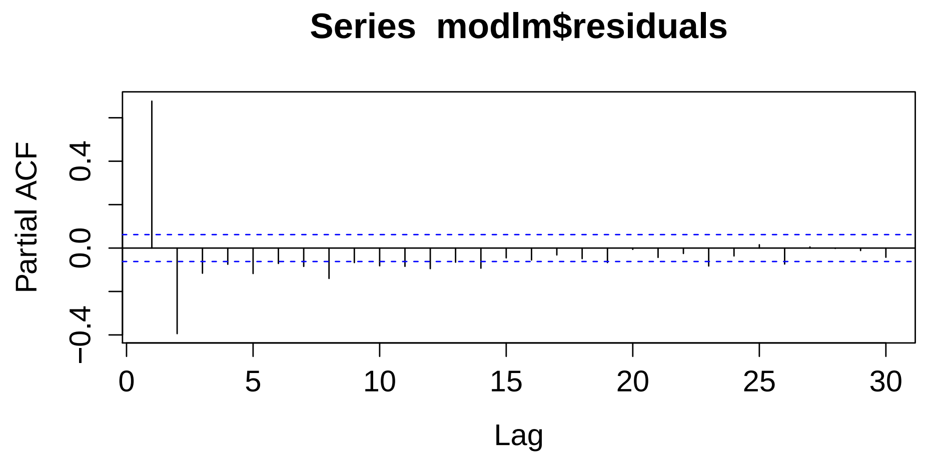


Figure 5: PACF of residuals

The pacf plot conforms that the process is of AR(2) model. Fitting an AR(2) model gives the coefficients as

arima(data1, order = c(2, 0, 0))

Coefficients:

	ar1	ar2
	-0.9434	-0.3940
s.e.	0.029	0.029

The ACF of residuals is shown in Figure 7

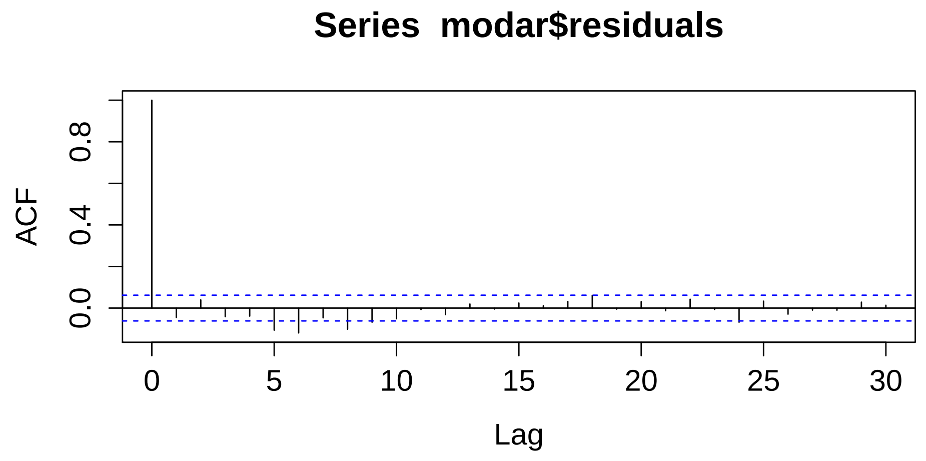


Figure 6: ACF of residuals

Clearly the ACF plot resembles like a white noise process. Hence the process is modelled as

$$x[k] = 2.22\sin(0.2\pi k) + v[k] \quad \text{where} \quad v[k] = -0.9434v[k-1] - 0.394v[k-2] + e[k]$$

(b).

The ACF of the data is shown in Figure 8

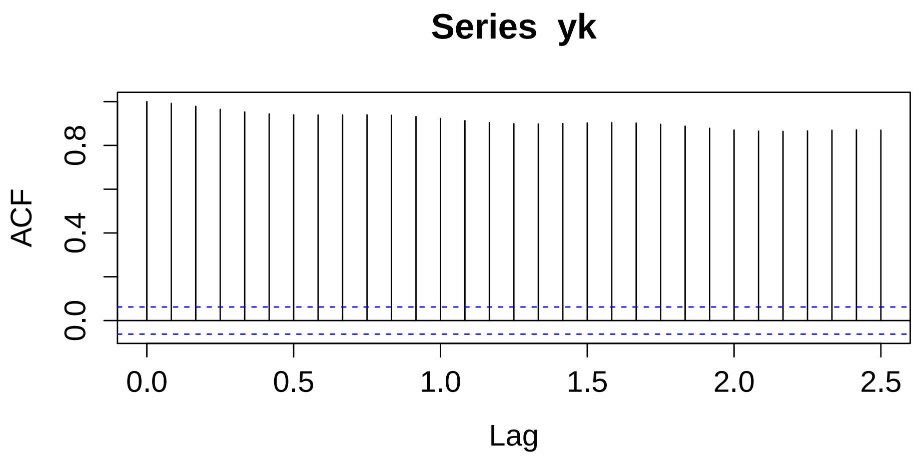


Figure 7: ACF of sarima data

Clearly, it shows non-stationarity. Hence the series is differenced and the acf of differenced series is shown in Figure 9

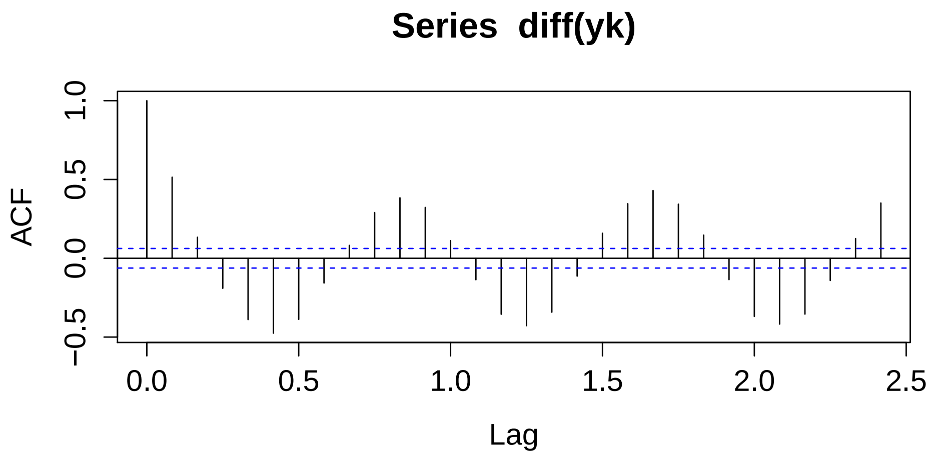


Figure 8: ACF of differenced series

It has seasonality of 10 samples. Hence the series is modelled using arima command in R as

```
mod=arima(yk,order=c(1,0,0),seasonal=list(period=10,order=c(1,0,1)))
```

Coefficients:

	ar1	ar2	sar1	sma1
	1.4971	-0.5041	-0.5242	0.6545
s.e.	0.0276	0.027	0.1119	0.0982

The acf plot of residuals is shown in Figure 10

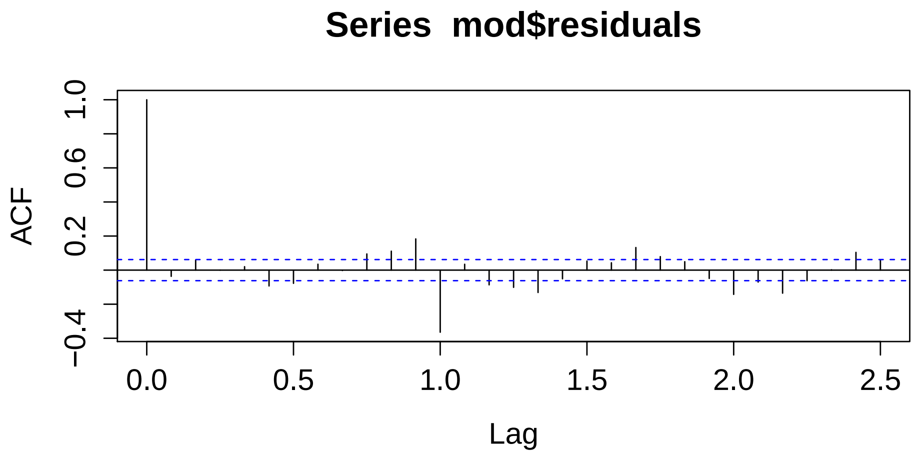


Figure 9: ACF of residuals

ACF plot almost resembles like that of white noise process. Hence the final model is

$$\hat{x}[k] = (1.3893x[k-1] - 0.557x[k-2]) \times (0.5242x[k-10] - 0.6545e[k-10])$$

4.

Simulated Series

The series is simulated with innovation variance 1. The time series plot is shown in Figure 10, while the ACF and PACF plots of the series are shown in Figure 11.

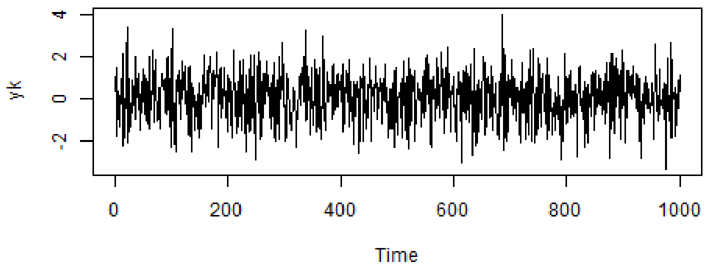


Figure 10: Time Series Plot

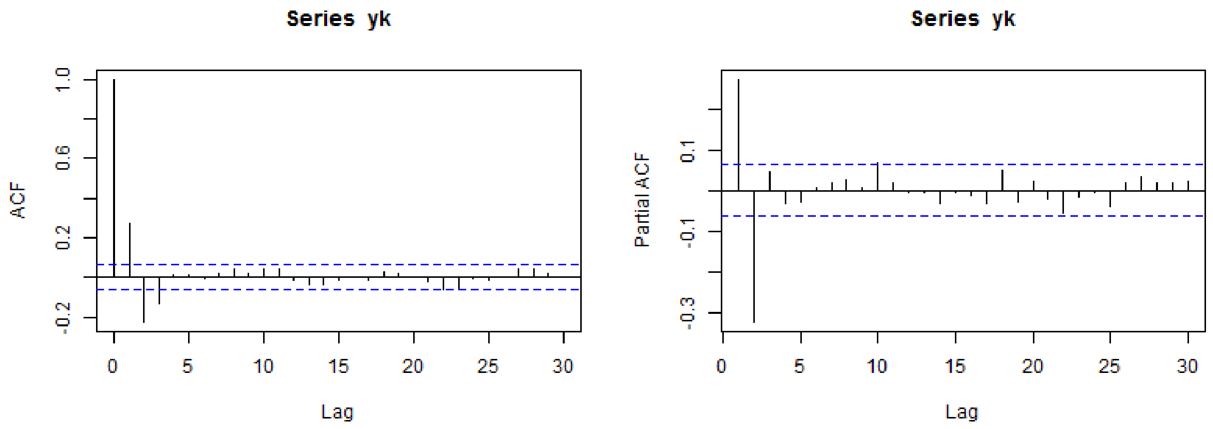


Figure 11: ACF and PACF of series

(a). Estimation from Wiener Khinchin Theorem

From the Wiener-Khinchin Theorem,

$$
\begin{aligned}
\gamma_{vv}(\omega) &= \frac{1}{2\pi}\sum_{l=-\infty}^{\infty} \sigma[l]e^{-j\omega l} \\
&= \frac{1}{2\pi}\left(\sigma[0] + 2\sum_{l=0}^{\infty}\sigma[l]cos(\omega l))\right)
\end{aligned}
$$

We usually do not have estimates of the ACVF at large lags. However, the ACVF of a stationary time-series dies down after a few lags. Hence, we can truncate the above expression to the first few lags.

$$\gamma_{vv}(\omega) = \frac{1}{2\pi}\left(\sigma[0] + 2\sum_{l=0}^{l_{max}} \sigma[l]cos(\omega l))\right)$$

From the ACF plot shown in Figure 11, the last significant lag is 3. We therefore truncate the expression to the first three lags. We also consider cases where the maximum lag is taken to be a higher value. The estimated PSD is shown in Figure 12.

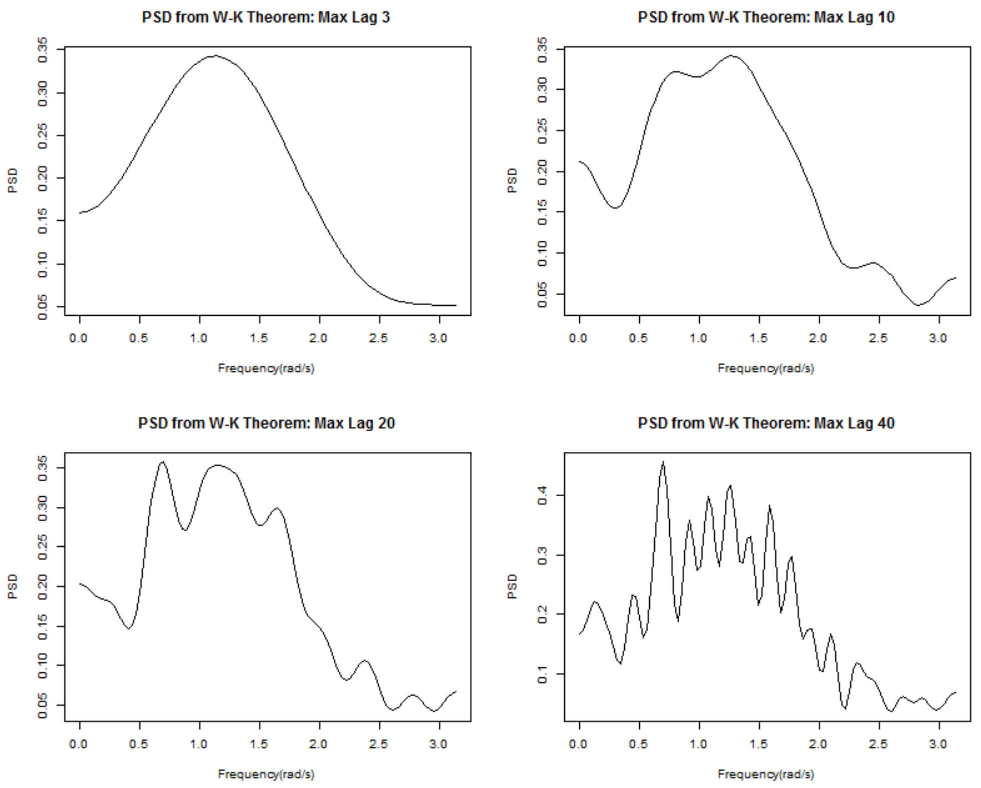


Figure 12: Estimated Spectral density from Wiener Khinchin Theorem

It is clear that the estimate is very sensitive to the maximum lag chosen, and it gets more noisy as the maximum lag is increased.

(b). Estimation from Time-series Model

From Figure 11, the ACF plot dies down after lag 3 while the PACF dies down after lag 2. Therefore as a first guess, we fit a ARMA(2,3) model. The fitted model is as follows:

```
Call:
arima(x = yk, order = c(2, 0, 3))
```

```
Coefficients:
coef ar1      ar2     ma1      ma2      ma3
val. 0.0981  -0.1250  0.2778  -0.1196  -0.0922
s.e. 0.2546   0.1752  0.2546   0.1318   0.1042

sigma^2 estimated as 1.002:  log likelihood = -1419.96,  aic = 2853.91
```

All the fitted coefficients are insignificant, which implies that the underlying series has white-noise characteristics. This is clearly not the case. Instead, we now fit an AR(2) model. The fitted model is as follows:

```
Call:
arima(x = yk, order = c(2, 0, 0))

Coefficients:
coef   ar1      ar2
val.   0.3602  -0.3233
s.e.   0.0299   0.0299

sigma^2 estimated as 1.005:  log likelihood = -1421.71,  aic = 2851.42
```

Here the coefficients are significant. The ACF and PACF plots for the model residuals, shown in Figure 13, indicate white-noise characteristics. Hence, the model is appropriate.

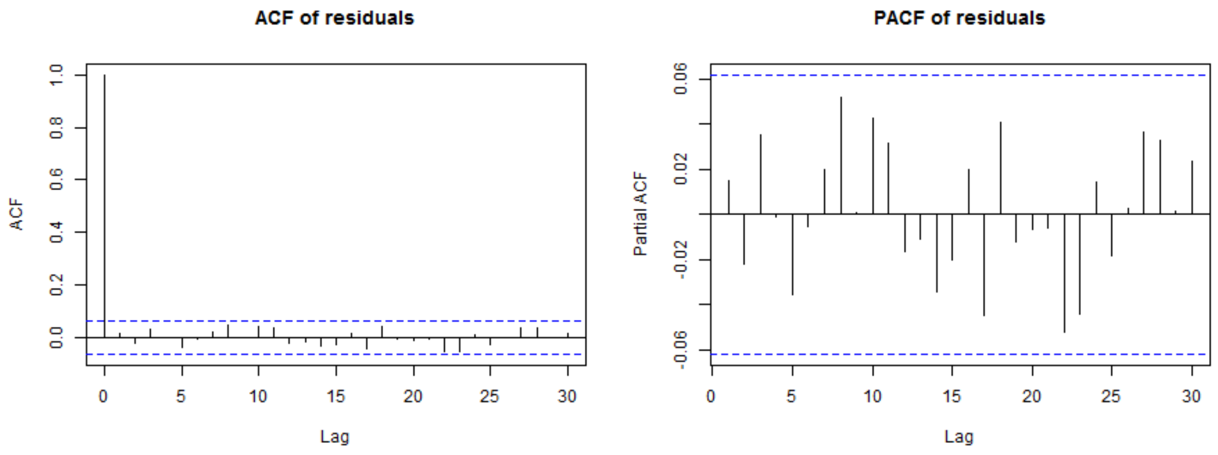


Figure 13: ACF and PACF of residuals of fitted model

The power-spectral density is estimated from the time-series model as follows:

$$\gamma_{vv}(\omega) = |\hat{H}(\omega)|^2 \frac{\hat{\sigma}_e^2}{2\pi}$$

where $\hat{H}(\omega)$ is the transfer-function form of the estimated model, and $\hat{\sigma}_e^2$ is the estimated variance.

Figure 14 shows the estimated spectral density plots. The plot on the left corresponds to the estimated model, while the plot on the right corresponds to the original model.

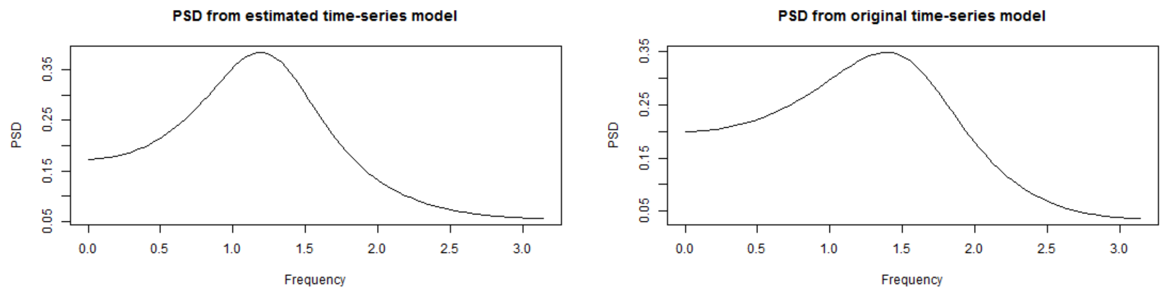


Figure 14: Estimated Spectral density from Time Series model

It is clear that there are some differences between these plots. This is because the fitted model is different from the actual one. Hence, the method can produce incorrect estimates depending on the model fit.

Evaluation of the methods

The first method is easier to use, since we do not have to fit any model (non-parametric approach). However, the estimate is sensitive to the number of lags used and hence it is noisy. On the other hand, we get smooth estimates from the time-series model (parametric) approach. This is because we have much less parameters to fit. However, the fitted model may differ from the data generating process and may result in incorrect estimates.

Code

```r
# Simulate Series
set.seed(323)
yk <- arima.sim(model=list(ar=c(0,-0.25),ma=0.4),1000)

## PSD from Weiner-Khinchin
```

```r
psd_from_acvf <- function(w,acvf,maxlag){
k <- 1:maxlag
psd <- (acvf[1] + 2*sum(acvf[-1]*cos(w*k)))/2/pi
psd
}

w <- seq(0,pi,length=100)
maxlag <- c(3,10,20,40)
gamma_wk <- matrix(nrow=100,ncol=4)

for(i in seq_along(maxlag)){
acvf <- acf(yk,maxlag[i],type = "covariance",plot = F)
gamma_wk[,i] <- sapply(w,psd_from_acvf,acvf$acf,maxlag[i])
}

## PSD from time-series modelling
mod1 <- arima(yk,order=c(2,0,3))
mod2 <- arima(yk,order=c(2,0,0))

# transfer function
H <- function(w) 1/(1-0.36*exp(-1i*w)+0.32*exp(-2i*w))
gamma_mod <- abs(sapply(w,H))^2*1.005/2/pi

# actual
Hact <- function(w) (1+0.4*exp(-1i*w))/(1+0.25*exp(-2i*w))
gamma_act <- abs(sapply(w,Hact))^2*1/2/pi
```

5.

Given exponential distribution

$$f(y) = \lambda e^{-\lambda y}$$

For the estimator to be efficient the function

$$\hat{\theta} = \frac{s(\theta)}{I(\theta)} + \theta$$

is to be independent of theta.

(a) $\hat{\theta} = \lambda$

The log-likelihood function is given by

$$L = \ln f(y) = \ln \lambda - \lambda y$$

Assuming the parameter to be estimated is $\theta = \lambda$, the score function is obtained as

$$S(\lambda) = \frac{\partial L}{\partial \theta} = \frac{1}{\lambda} - y$$

The information matrix is obtained as

$$\begin{aligned} I(\lambda) &= -E(\frac{\partial^2 L}{\partial \lambda^2}) \\ &= \frac{1}{\lambda^2} \end{aligned}$$

Here $\hat{\theta}$ is obtained as

$$\hat{\theta} = \lambda^2 y$$

which is a function of λ. Hence there exists no efficient estimator to estimate λ.

(b) $\hat{\theta} = \frac{1}{\lambda}$

The log-likelihood function is given by

$$L = \ln f(y) = \ln \lambda - \lambda y$$

Assuming the parameter to be estimated is $\theta = \frac{1}{\lambda}$, the score function is obtained as

$$S(\frac{1}{\lambda}) = \frac{\partial L}{\partial \theta} = -\lambda + \lambda^2 y$$

The information matrix is obtained as

$$\begin{aligned} I(\lambda) &= -E(\frac{\partial^2 L}{\partial \lambda^2}) \\ &= \lambda^2 \end{aligned}$$

Here $\hat{\theta}$ is obtained as

$$\hat{\theta} = y$$

which is a not a function of $\frac{1}{\lambda}$. Hence there exists an efficient estimator to estimate $\frac{1}{\lambda}$.

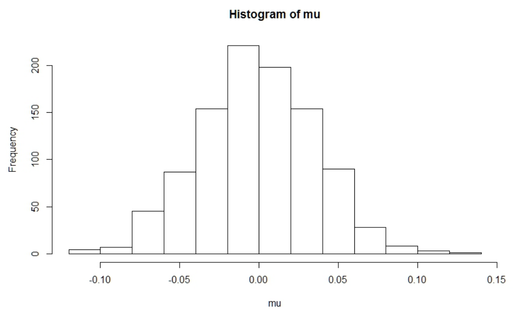
6.

Let $X_1,\ X_2,\ X_3 ... X_N$ be the set of N samples of a stationary process with mean μ.
Therefore mean of the sample mean. $\Rightarrow E(\frac{(X_1+X_2+X_3....X_N)}{N}) \Rightarrow \mu$
$\bar{X}_i$ be the sample mean of the i^{th} observation and is given by $\frac{(X_1+X_2+X_3....X_N)}{N}$
Variance of the sample mean $var(\bar{X}) = E(\bar{X}_i - \mu)^2$

$$var(\bar{X}) = E(\frac{(X_1+X_2+X_3....X_N)^2}{N^2} - \mu^2)$$
$$\Rightarrow \frac{1}{N^2}(E(X_1^2 + X_2^2 + X_3^2 + ...X_N^2 + 2\sum(X_iX_j) - N\mu^2 + N\mu^2) + \mu^2$$
$$\Rightarrow \frac{1}{N^2}(E(X_1^2) - \mu^2 + E(X_2^2) - \mu^2 + E(X_3^2) - \mu^2 + ...E(X_N^2) - \mu^2 + 2\sum E(X_iX_j) + N\mu^2) + \mu^2$$
$$\Rightarrow \frac{1}{N^2}(N\sigma[0] + 2\sum E(X_iX_j) + N\mu^2) + \mu^2 \Rightarrow \frac{1}{N}\sigma[0] + 2\frac{\sum E(X_iX_j)}{N^2} + \frac{\mu^2}{N} + \mu^2$$
$$\Rightarrow \frac{1}{N}[\sigma[0] + 2\sum_{1}^{N-1}(1 - \frac{|l|}{N})\sigma[l]]$$

Verifying the above result with the MA(1) process by R $x[k] = e[k] + 0.4e[k-1]$

```
N = 1000
Ns = 1500
mu = array(0, dim=c(1,N))
for (i in 1:N){
vk <- arima.sim(model = list(ma = 0.4, order = c(0,0,1)), n = Ns)
mu[i] = mean(vk)
}
mean_mu <- mean(mu)
hist(mu)
```



Comparing the variances obtained in R

```
var1_mu = 0
for (j in 1:N){
var1_mu = var1_mu + mu[j]-mean_mu
}
var1_mu = var1_mu/(N-1)

acvf_vk <- acf(vk)
acvf_vk <- acf(vk, lag.max = Ns, type = "cov")
acvf_vk <- acvf_vk$acf

s = 0
for (k in 2:Ns){
s = s +(1-k/Ns)*acvf_vk[k]
}

var2_mu <- 1/Ns*(acvf_vk[1]+2*s)

c(var1_mu, var2_mu)
[1] 3.184668e-18 4.073989e-04

```